



# **Local features, distances and kernels**

**February 5, 2009**

**Kristen Grauman  
UT-Austin**

## **Plan for today**

- **Lecture** : local features and matching
- **Papers**:
  - Video Google [Sivic & Zisserman]
  - Pyramid match [Grauman & Darrell]
  - Learning local distance functions [Frome et al.]
- **Demo**:
  - Feature sampling strategies for categorization

## Local features: motivation



- Last week: appearance-based features assuming window under consideration
  - Is fairly aligned across examples
  - Has similar total structure, same components present
- This week: local representations to offer robustness to occlusion, clutter, viewpoint changes,...
  - How to describe
  - How to compare

## Local invariant features

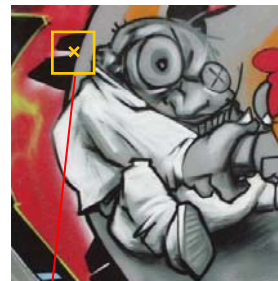
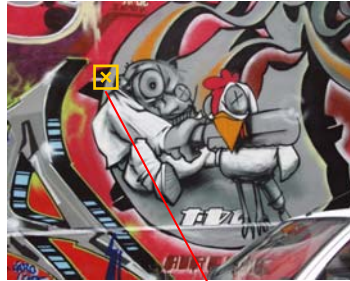
- Problem 1:
  - Detect the *same point independently* in both images



no chance to match!

We need a repeatable detector

## Automatic Scale Selection



$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

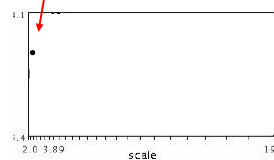
Same operator responses if the patch contains the same image up to scale factor

How to find corresponding patch sizes?

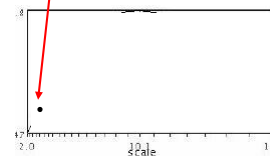
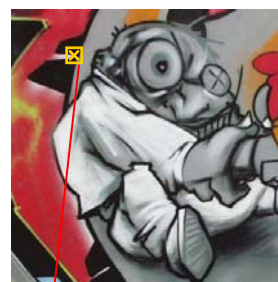
Slide credit K. Grauman, B. Leibe AAAI08 Short Course

## Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$$f(I_{i_1 \dots i_m}(x, \sigma))$$

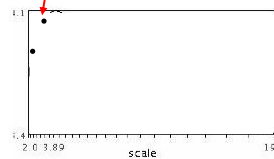


$$f(I_{i_1 \dots i_m}(x', \sigma))$$

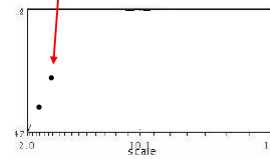
Slide credit K. Grauman, B. Leibe AAAI08 Short Course

## Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$$f(I_{i_1...i_m}(x, \sigma))$$

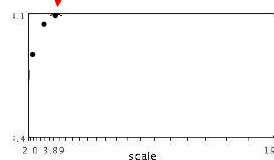
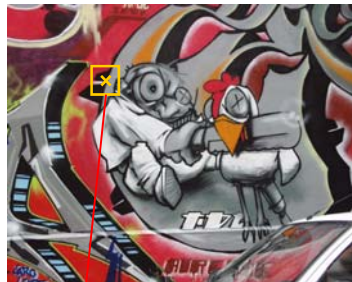


$$f(I_{i_1...i_m}(x', \sigma))$$

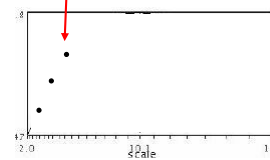
Slide credit K. Grauman, B. Leibe AAAI08 Short Course

## Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$$f(I_{i_1...i_m}(x, \sigma))$$

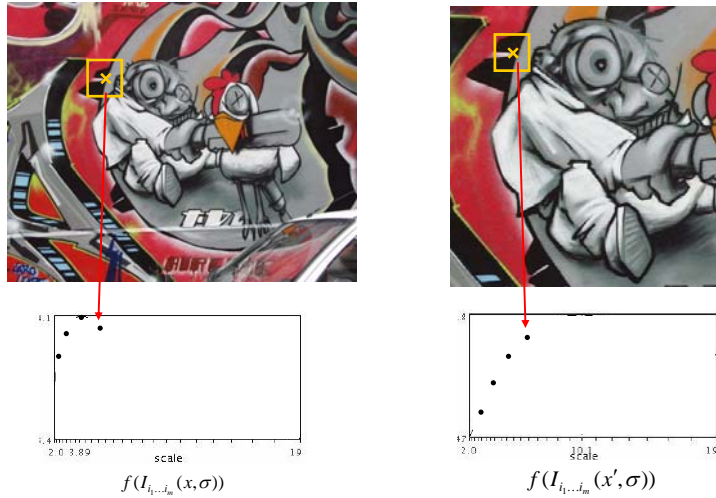


$$f(I_{i_1...i_m}(x', \sigma))$$

Slide credit K. Grauman, B. Leibe AAAI08 Short Course

## Automatic Scale Selection

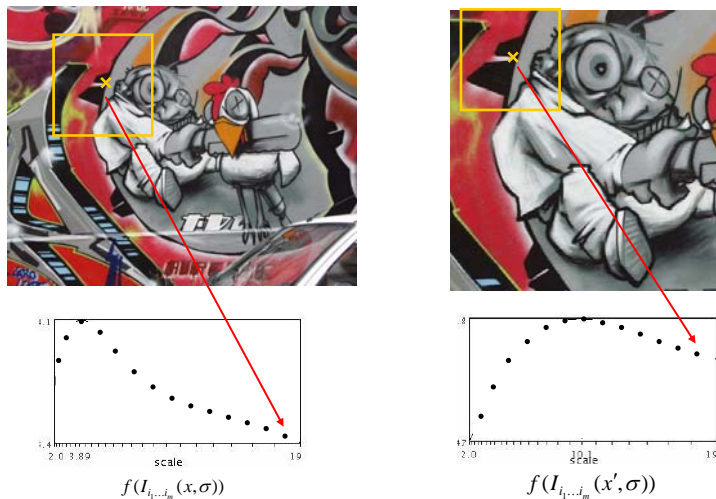
- Function responses for increasing scale (scale signature)



Slide credit K. Grauman, B. Leibe AAAI08 Short Course

## Automatic Scale Selection

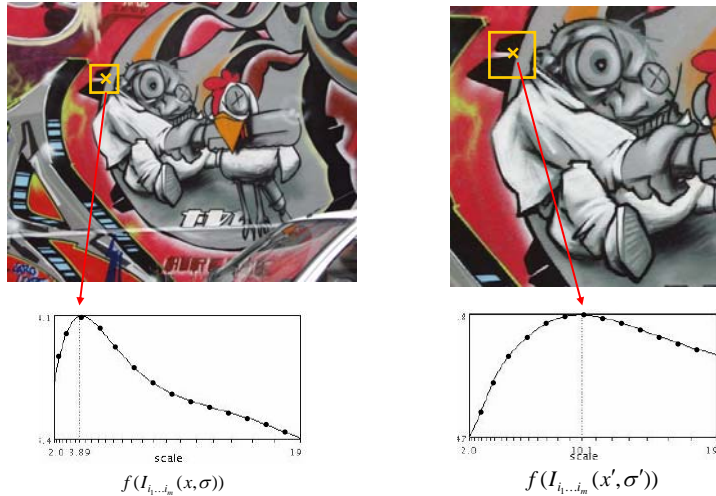
- Function responses for increasing scale (scale signature)



Slide credit K. Grauman, B. Leibe AAAI08 Short Course

## Automatic Scale Selection

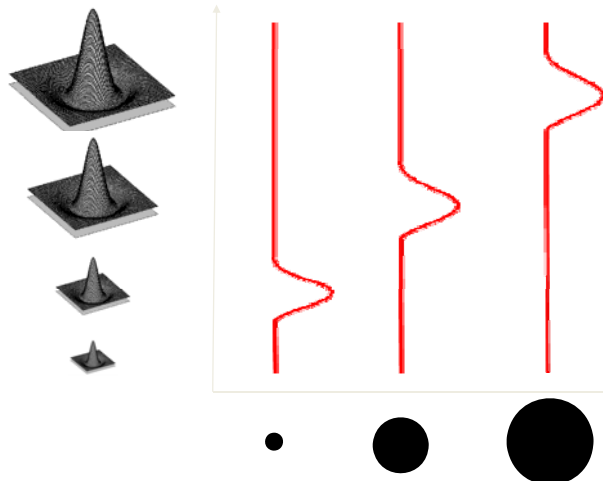
- Function responses for increasing scale (scale signature)



Slide credit K. Grauman, B. Leibe AAAI08 Short Course

## What Is A Useful Signature Function?

- Laplacian-of-Gaussian = “blob” detector



Slide credit K. Grauman, B. Leibe AAAI08 Short Course



## Scale-space blob detector: Example



Source: Lana Lazebnik

## Scale-space blob detector: Example



sigma = 11.9912

Source: Lana Lazebnik

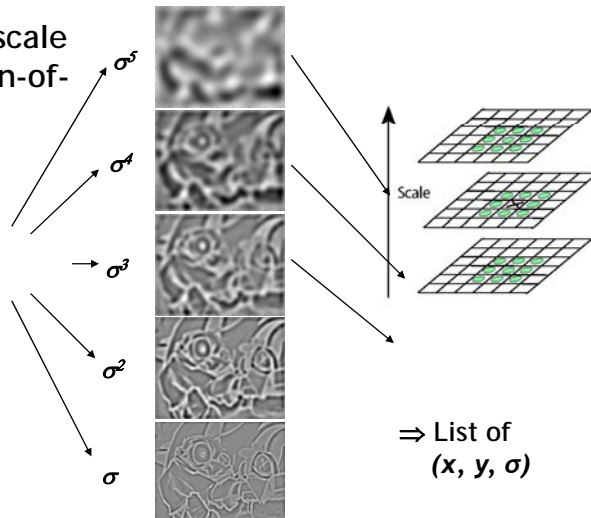
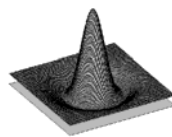
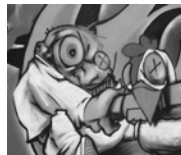
## Scale-space blob detector: Example



Source: Lana Lazebnik

## Laplacian-of-Gaussian (LoG) for scale invariant detection

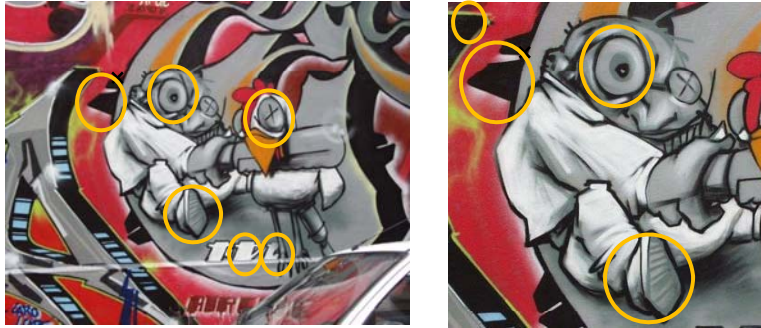
- Local maxima in scale space of Laplacian-of-Gaussian



Slide credit K. Grauman, B. Leibe AAAI08 Short Course

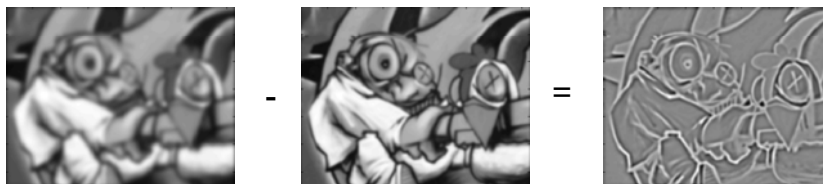
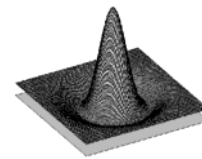


## Laplacian of Gaussian: scale invariant detection



## Difference-of-Gaussian (DoG)

- Difference of Gaussians gives an efficient approximation of the Laplacian-of-Gaussian



Slide credit K. Grauman, B. Leibe AAAI08 Short Course

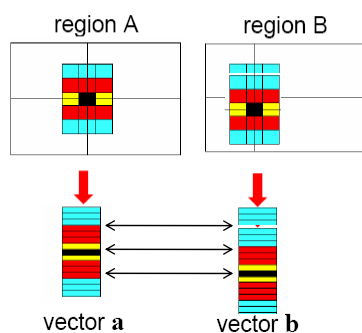
## Local invariant features

- Problem 2:
  - For each point correctly recognize the corresponding one



We need a reliable and distinctive descriptor

## Raw patches as local descriptors



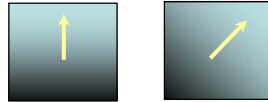
The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.

## Rotation invariant descriptors

- Find local orientation

Dominant direction of gradient for the image patch



- Rotate patch according to this angle

This puts the patches into a canonical orientation.

What about illumination and translation?

## SIFT Descriptor [Lowe 2004]

- Use histograms to bin pixels within sub-patches according to their orientation.
- $4 \times 4 \times 8 = 128$  dimensional feature vector

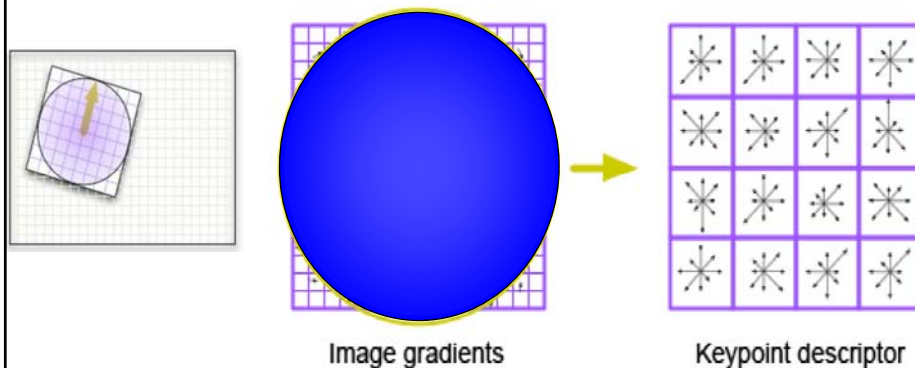


Image from: Jonas Hurreimann Slide credit: O. Pele, S. Thrun, J. Kořecká, N. Kumar

## SIFT Descriptor [Lowe 2004]

Extraordinarily robust matching technique

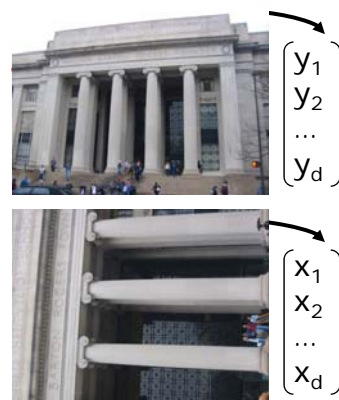
- Can handle changes in viewpoint
  - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Lots of code available
  - [http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known\\_implementations\\_of\\_SIFT](http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT)



## Local invariant features: basic flow

- 1) Detect interest points
- 2) Extract descriptors

Descriptors map each region in image to a (typically high-dimensional) feature vector.



## Local representations

Many options for detection & description...



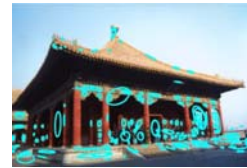
SIFT [Lowe 99]



Shape context  
[Belongie 02]



Superpixels  
[Ren et al.]



Maximally Stable  
Extremal Regions  
[Matas 02]



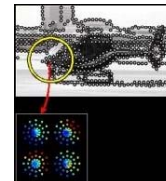
Salient regions  
<sup>25</sup> [Kadir 01]



Harris-Affine  
[Mikolajczyk 04]



Spin images  
[Johnson 99]




Geometric Blur  
[Berg 05]



## You Can Try It At Home...

- For most local feature detectors, executables are available online:
- <http://robots.ox.ac.uk/~vgg/research/affine>
- <http://www.cs.ubc.ca/~lowe/keypoints/>
- <http://www.vision.ee.ethz.ch/~surf>

## Affine Covariant Features




KATHOLIEKE UNIVERSITEIT  
**LEUVEN**

This work is only related to the Visual Geometry Group, Katholieke Universiteit Leuven, Intel Research and the Center for Machine Perception.

## Affine Covariant Region Detectors

Input image



Region detector

→


Detector output

```
format:
1.0
m
u1 v1 a1 b1 c1
.
.
um vm am bm cm
output example:
img1.haraff
```

display features.m

→

Image with displayed regions



**Parameters defining an affine region**  
 $u, v, a, b, c$  in  $a(x-u) + 2b(x-u)(y-v) + c(y-v)^2 = 1$   
 with  $(0, 0)$  at image top left corner

**Code**  
 - provided by the authors, see [publications](#) for details and links to authors web sites.

Linux binaries	Example of use	Displaying
<a href="#">Harris-Affine &amp; Hessian-Affine</a>	<code>prompt&gt; ./h_affine.in -haraff -1 <a href="#">img1.ppm</a> -o img1.haraff -thres 1000</code>	<code>matlab&gt;&gt; ;</code>
	<code>prompt&gt; ./h_affine.in -hesaff -1 <a href="#">img1.ppm</a> -o img1.hesaff -thres 500</code>	<code>matlab&gt;&gt; ;</code>
<a href="#">MSER</a> - Maximally stable extremal regions (also Windows)	<code>prompt&gt; ./msr.in -t 2 -es 2 -1 <a href="#">img1.ppm</a> -o img1.mser</code>	<code>matlab&gt;&gt; ;</code>
<a href="#">IRF</a> - Intensity extrema based detector	<code>prompt&gt; ./ifr.in <a href="#">img1.ppm</a> img1.ifr -scalefactor 1.0</code>	<code>matlab&gt;&gt; ;</code>
<a href="#">EBF</a> - Edge based detector	<code>prompt&gt; ./ebr.in <a href="#">img1.ppm</a> img1.ebr</code>	<code>matlab&gt;&gt; ;</code>
<a href="#">Salient</a> region detector	<code>prompt&gt; ./salient.in <a href="#">img1.ppm</a> img1.sal</code>	<code>matlab&gt;&gt; ;</code>

<http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries>

## Applications of local invariant features & matching

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
  - Specific objects
  - Textures
  - Categories
- ...



## Wide baseline stereo



[Image from T. Tuytelaars ECCV 2006 tutorial]

## Panorama stitching



(a) Matier data set (7 images)



(b) Matier final stitch

Brown, Szeliski, and Winder, 2005

## Automatic mosaicing



<http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html>

## Recognition of specific objects, scenes



Schmid and Mohr 1997



Sivic and Zisserman, 2003



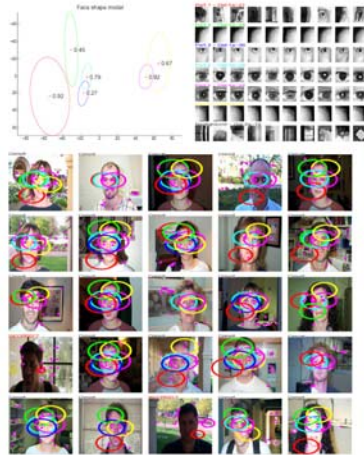
Rothganger et al. 2003



Lowe 2002

## Recognition of categories

Constellation model



Weber et al. (2000)  
Fergus et al. (2003)

Bags of words

Database	Sample cluster #1	Sample cluster #2
Airplanes		
Motorbikes		
Leaves		
Wild Cats		
Faces		
Bicycles		
People		

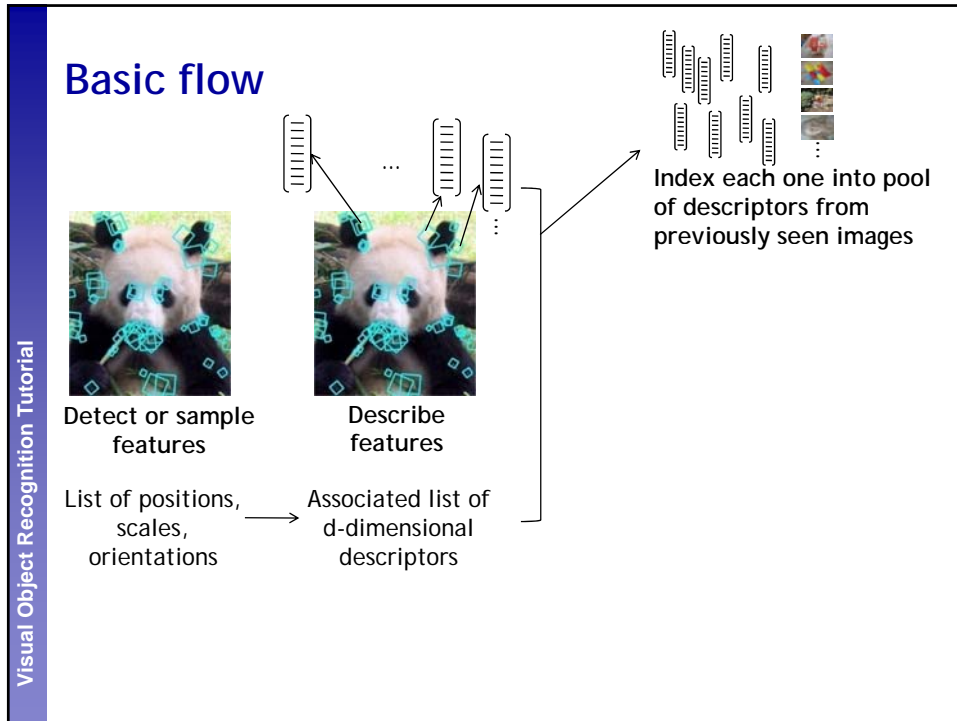
Csurka et al. (2004)  
Dorko & Schmid (2005)  
Sivic et al. (2005)  
Lazebnik et al. (2006), ...

[Slide from Lana Lazebnik, Sicily 2006]

## Value of local features

- Critical to find distinctive and repeatable local regions for multi-view matching
- Complexity reduction via selection of distinctive points
- Describe images, objects, parts without requiring segmentation; robustness to clutter & occlusion
- Robustness: similar descriptors in spite of moderate view changes, noise, blur, etc.

*Once we have the features themselves, how to use for recognition, search?*



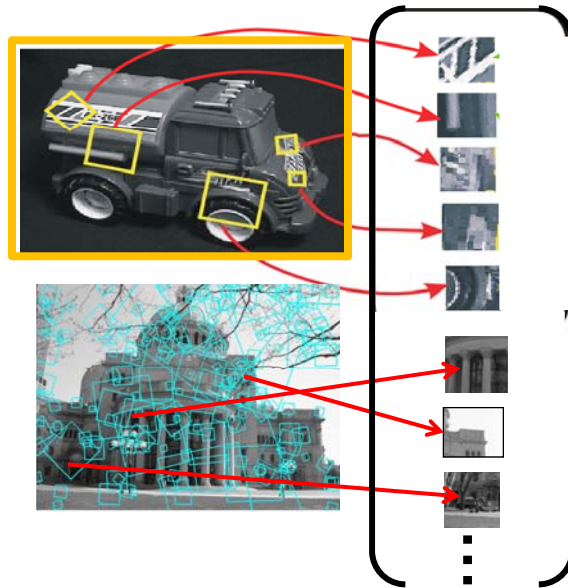
## Pose clustering and verification with SIFT

To detect **instances** of objects from a model base:



- 1) Index descriptors (distinctive features narrow possible matches)

## Indexing local features




## Pose clustering and verification with SIFT

To detect **instances** of objects from a model base:




- 1) Index descriptors (distinctive features narrow possible matches)
- 2) Generalized Hough transform to vote for poses (keypoints have record of parameters relative to model coordinate system) *[next week]*
- 3) Affine fit to check for agreement between model and image features (approximates perspective projection for planar objects)

## Planar objects




## Model images and their SIFT keypoints



## Input image

Model keypoints that were used to recognize, get least squares solution.



## Recognition result

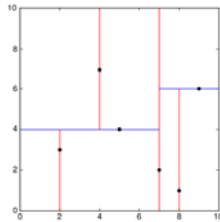
[Lowe]

## Indexing local features

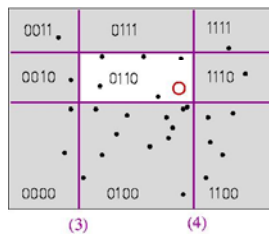
- With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?
  - Low-dimensional descriptors : can use standard efficient data structures for nearest neighbor search
  - High-dimensional descriptors: approximate nearest neighbor search methods more practical
  - Inverted file indexing schemes



## Indexing local features: approximate nearest neighbor search



Best-Bin First (BBF), a variant of k-d trees that uses priority queue to examine most promising branches first [Beis & Lowe, CVPR 1997]



(1) Locality-Sensitive Hashing (LSH), a randomized hashing technique using hash functions that map similar points to the same bin, with high probability [Indyk & Motwani, 1998]

(2)

(3)

(4)

41

## Indexing local features

- With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?
    - Low-dimensional descriptors : can use standard efficient data structures for nearest neighbor search
    - High-dimensional descriptors: approximate nearest neighbor search methods more practical
- Inverted file indexing schemes

K. Grauman, B. Leibe

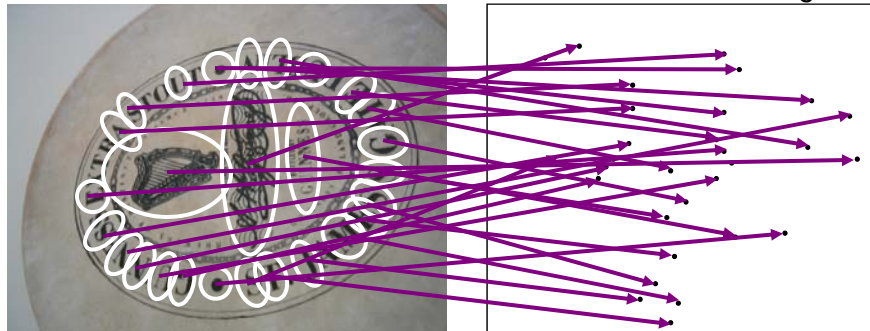
## Indexing local features: inverted file index

Index		
"Along I-75," From Detroit to Florida; inside back cover	Butterfly Center, McGuire; 134	Driving Lanes; 85
"Drive I-95," From Boston to Florida; inside back cover	CAA (see AAA)	Duval County; 163
1929 Spanish Trail Roadway; 101-102, 104	CCC, The; 111, 113, 115, 135, 142	Eau Gallie; 175
511 Traffic Information; 83	Ca #Zan; 147	Edison, Thomas; 152
A1A (Barrier Is.) - I-95 Access; 86	Caloosahatchee River; 152	Eglin AFB; 116-118
AAA (and CAA); 83	Name; 150	Eight Roads; 176
AAA National Office; 88	Canaveral Nat'l Seashore; 173	Ellenton; 144-145
Abbreviations;	Cannon Creek Airport; 130	Emanuel Point Wreck; 120
Colored 25 mile Maps; cover	Canopy Road; 106, 109	Emergency Callboxes; 83
Exit Services; 196	Cape Canaveral; 174	Epiphytes; 142, 148, 157, 159
Towtigue; 85	Castillo San Marcos; 169	Escambia Bay; 119
Africa; 177	Cave Diving; 131	Bridge (I-10); 119
Agricultural Inspection Shts; 126	Cayo Costa, Name; 150	County; 120
Ar-Tah-The-Ki Museum; 180	Celebration; 99	Edney; 153
Air Conditioning, First; 112	Charlotte County; 149	Everglades; 90, 95, 139-140, 154-160
Alabama; 124	Charlotte Harbor; 150	Dwelling at; 156, 161
Alachua; 132	Chautauqui; 116	Wildlife; 160
County; 131	Chapley; 114	Wonder Gardens; 154
Alafia River; 143	Name; 115	Falling Waters SP; 115
Alapaha, Name; 126	Choctawhatchee, Name; 115	Fantasy of Flight; 95
Alfred B. Mackay Gardens; 106	Cirque Museum, Ringling; 147	Fayer Dykes SP; 171
Alligator Alley; 154-155	Citrus; 88, 97, 130, 136, 140, 180	Fires, Forest; 166
Alligator Farm, St. Augustine; 169	CityPlace, W. Palm Beach; 180	Fires, Prescribed; 148
Alligator Hole (redaction); 157	City Maps	Fisherman's Village; 151
Alligator, Buddy; 155	FL Leewardale Expressway; 194-195	Flagler County; 171
Alligators; 100, 125, 136, 147, 156	Jacksonville; 169	Flagler, Henry; 97, 165, 167, 171
Anastasia Island; 170	Kissimmee Expressway; 192-193	Florida Aquarium; 185
Amnack; 109-109, 146	Miami Expressways; 154-155	Florida;
Apalachicola River; 112	Olando Expressways; 192-193	12,000 years ago; 187
Appleton Mus of Art; 136	Pensacola; 26	Cavern SP; 114
Aquifer; 102	Tallahassee; 191	Map of all expressways; 2-3
Arabian Nights; 94	Tampa-St. Petersburg; 63	Mus of Natural History; 184
Art Museum, Ringling; 147	St. Augustine; 191	National Cemetery; 141
Aruba Beach Caks; 153	Civil War; 100, 102, 127, 138, 141	Part of Africa; 177
Aucilla River Project; 106	Cleaver Marine Aquarium; 187	Platform; 187
Babcock-Wheeler WMA; 151	Collier County; 154	Sherriff's Boys Camp; 126
Bahia Mar Marina; 184	Collier, Barron; 152	Sports Hall of Fame; 139
Baker County; 99	Colonial Spanish Quarters; 168	Sun 'n Fun Museum; 97
Bandford Mallen; 182	Columbia County; 101, 128	Supreme Court; 107
Barge Canal; 137	Cookins Building Material; 165	Florida's Turnpike (FTT); 178, 189
Bee Line E-90; 80	Corkscrew Swamp, Name; 154	25 mile Strip Maps; 66
Belt Cutlet Mark; 89	Cowboys; 85	Administration; 189
Bernard Castro; 136	Crab Trap II; 144	Coin System; 190
Big "Y"; 165	Cracker, Florida; 88, 95, 132	Exit Services; 189
Big Cypress; 155, 158	Crossedown Expo; 11, 35, 38, 143	H&T; 76, 161, 190
Big Foot Monster; 105	Cuban Bread; 184	History; 189
Billie Swamp Safari; 150	Dade Battleground; 140	Names; 189
Blackwater River SP; 117	Dade, Maj. Francis; 139-140, 161	Service Plazas; 190
Blue Angels	Dania Beach Hurricane; 184	Spur SR91; 76
AA-C Skyhawk; 117	Daniel Boone, Florida Walk; 117	Ticket System; 190
Altium; 121	Daytona Beach; 172-173	Toll Plazas; 190
Bike Routes; 22-27	De Land; 87	Ford, Henry; 152
	De Soto, Hernando,	Fort Barrancas; 122
	Anhaica; 108-109, 146	Buried Alive; 123
	County; 146	and Cuckoo; 184

- For text documents, an efficient way to find all **pages** on which a **word** occurs is to use an index...
- We want to find all **images** in which a **feature** occurs.
- To use this idea, we'll need to map our features to "visual words".

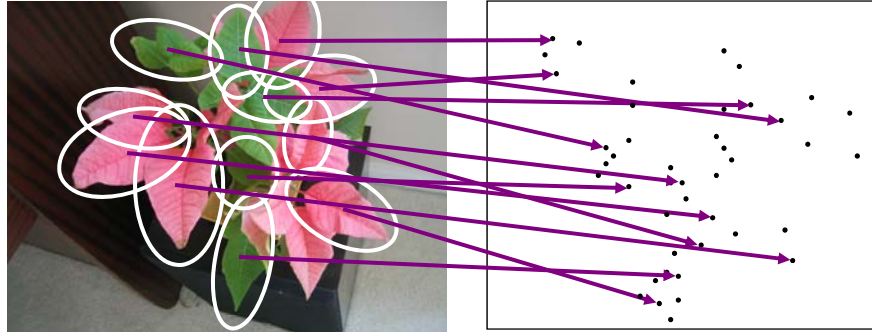
## Visual words: main idea

- Extract some local features from a number of images ...



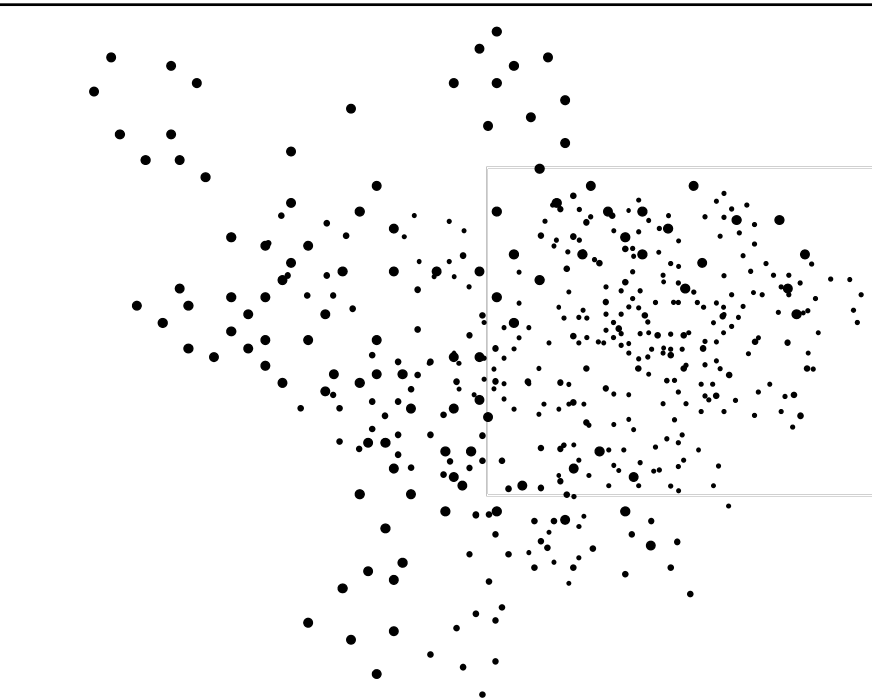
e.g., SIFT descriptor space: each point is 128-dimensional

## Visual words: main idea



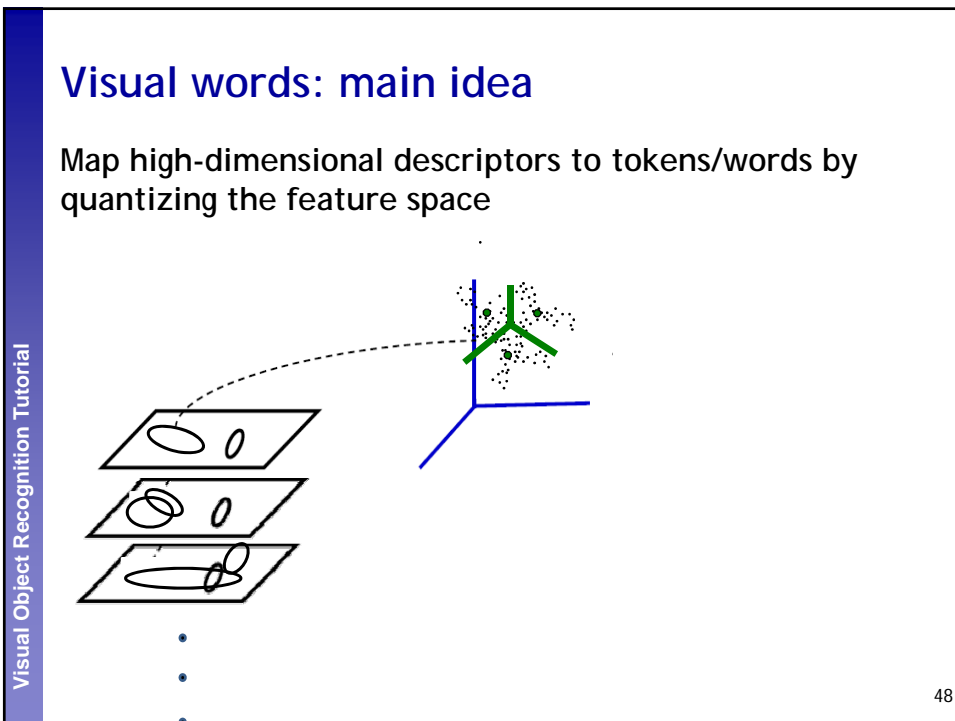
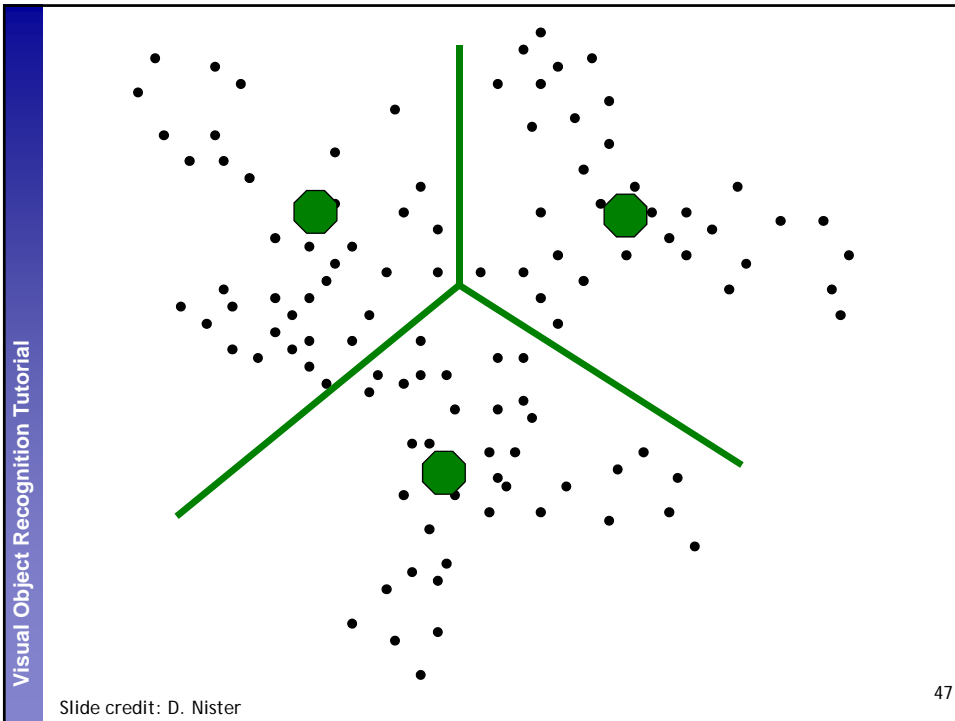
Slide credit: D. Nister

45



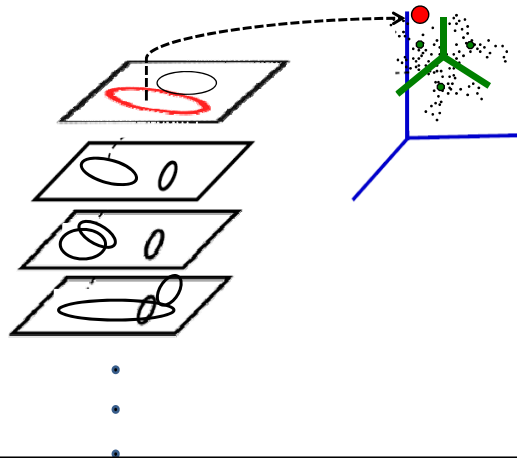
Slide credit: D. Nister

46



## Visual words: main idea

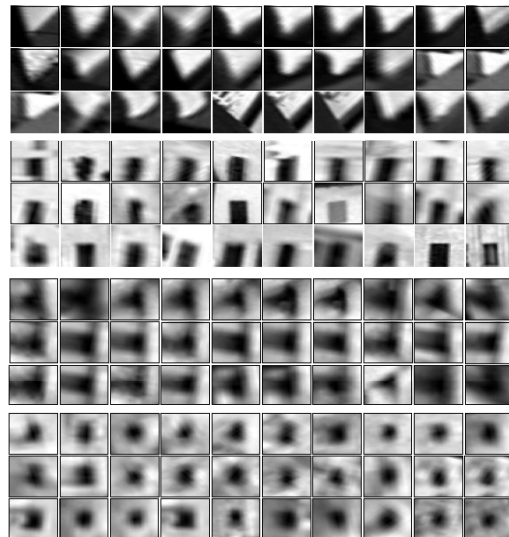
Map high-dimensional descriptors to tokens/words by quantizing the feature space



49

## Visual words

- Example: each group of patches belongs to the same visual word



## Inverted file index for images comprised of visual words



frame #5



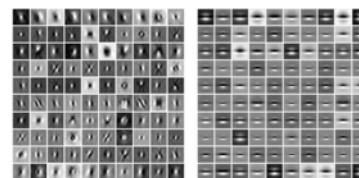
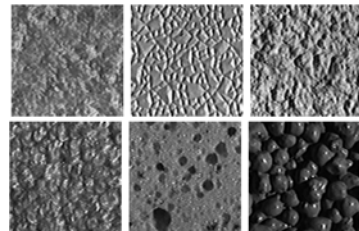
frame #10

Word number	List of image numbers
1	→ 5, 10, ...
2	→ 10, ...
...	...

Image credit: A. Zisserman

## Visual words

- First explored for texture and material representations
- **Texton** = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.

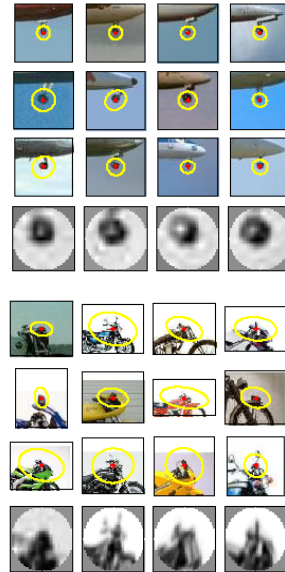


Leung & Malik 1999; Varma & Zisserman, 2002; Lazebnik, Schmid & Ponce, 2003;



## Visual words

- More recently used for describing scenes and objects for the sake of indexing or classification.



Sivic & Zisserman 2003; Csurka, Bray, Dance, & Fan 2004; many others.

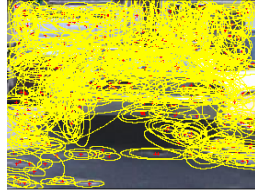
53

## Visual vocabulary formation

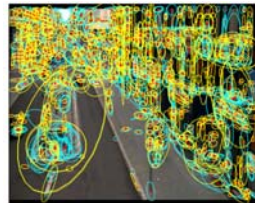
Issues:

- Sampling strategy: where to extract features?
- Clustering / quantization algorithm
- Unsupervised vs. supervised
- What corpus provides features (universal vocabulary?)
- Vocabulary size, number of words

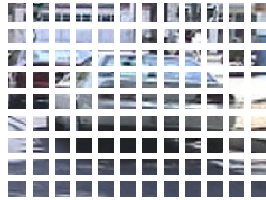
## Sampling strategies



Sparse, at  
interest points



Multiple interest  
operators



Dense, uniformly



Randomly

- To find specific, textured objects, sparse sampling from interest points often more reliable.
- Multiple complementary interest operators offer more image coverage.
- For object categorization, dense sampling offers better coverage.

See [Nowak, Jurie & Triggs, ECCV 2006] , and Gautam's demo!

Image credits: F-F. Li, E. Nowak, J. Sivic

## Clustering / quantization methods

- k-means (typical choice), agglomerative clustering, mean-shift,...
- Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies
  - Vocabulary tree [Nister & Stewenius, CVPR 2006]

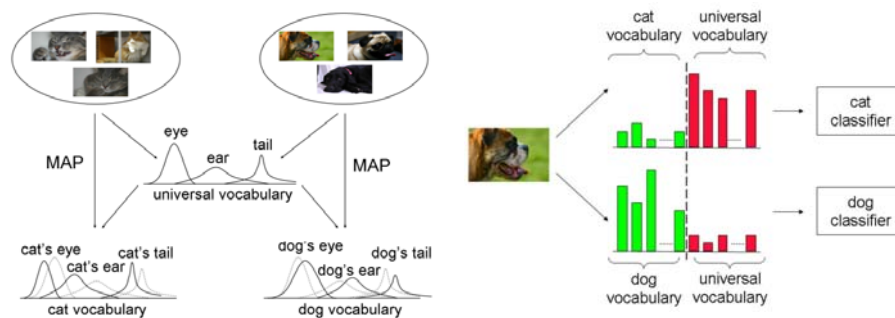
## Visual vocabulary formation

Issues:

- Sampling strategy: where to extract features?
- Clustering / quantization algorithm
- **Unsupervised vs. supervised**
- What corpus provides features (universal vocabulary?)
- Vocabulary size, number of words

## Supervised vocabulary formation

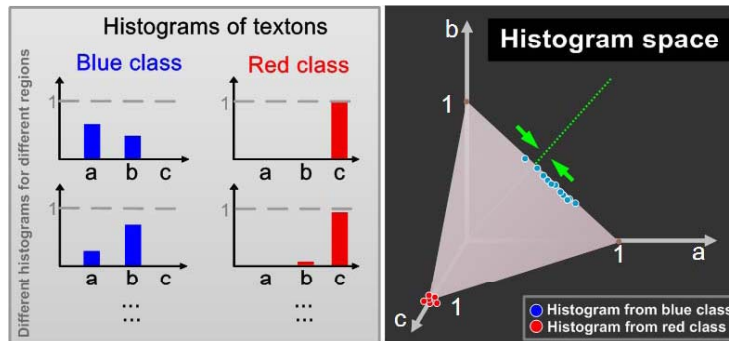
- Recent work considers how to leverage labeled images when constructing the vocabulary



Perronnin, Dance, Csaruka, & Bressan, Adapted Vocabularies for Generic Visual Categorization, ECCV 2006.

## Supervised vocabulary formation

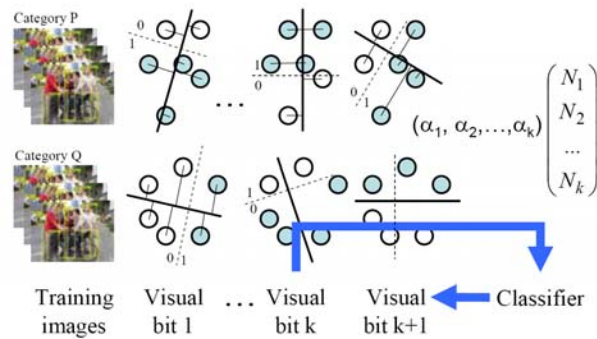
- Merge words that don't aid in discriminability



Winn, Criminisi, & Minka, Object Categorization by Learned Universal Visual Dictionary, ICCV 2005

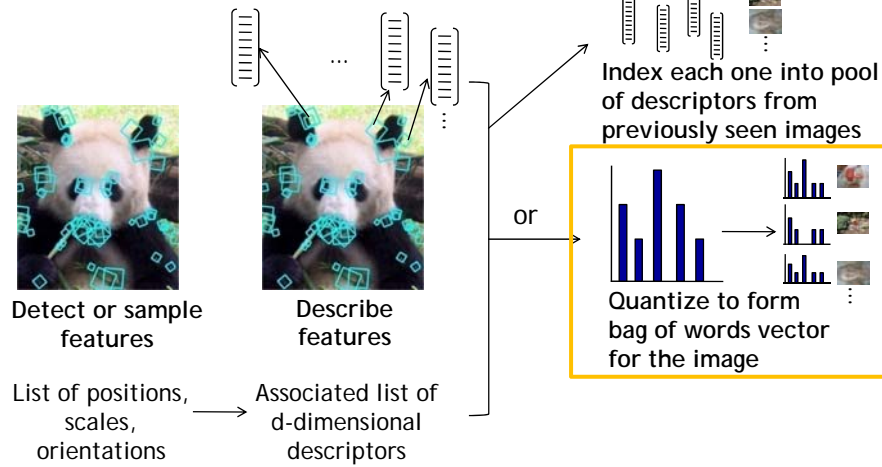
## Supervised vocabulary formation

- Consider vocabulary and classifier construction jointly.



Yang, Jin, Sukthankar, & Jurie, Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition, CVPR 2008.

## Basic flow



## Bags of visual words

- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.
- Set of patches -> vector
- Good empirical results for image classification.

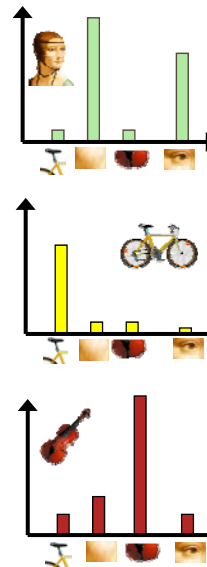


Image credit: Fei-Fei Li

## Bags of visual words for image recognition

Caltech 6 dataset



class	bag of features	bag of features	Parts-and-shape model
	Zhang et al. (2005)	Willamowski et al. (2004)	Fergus et al. (2003)
airplanes	<b>98.8</b>	97.1	90.2
cars (rear)	98.3	<b>98.6</b>	90.3
cars (side)	<b>95.0</b>	87.3	88.5
faces	<b>100</b>	99.3	96.4
motorbikes	<b>98.5</b>	98.0	92.5
spotted cats	<b>97.0</b>	—	90.0

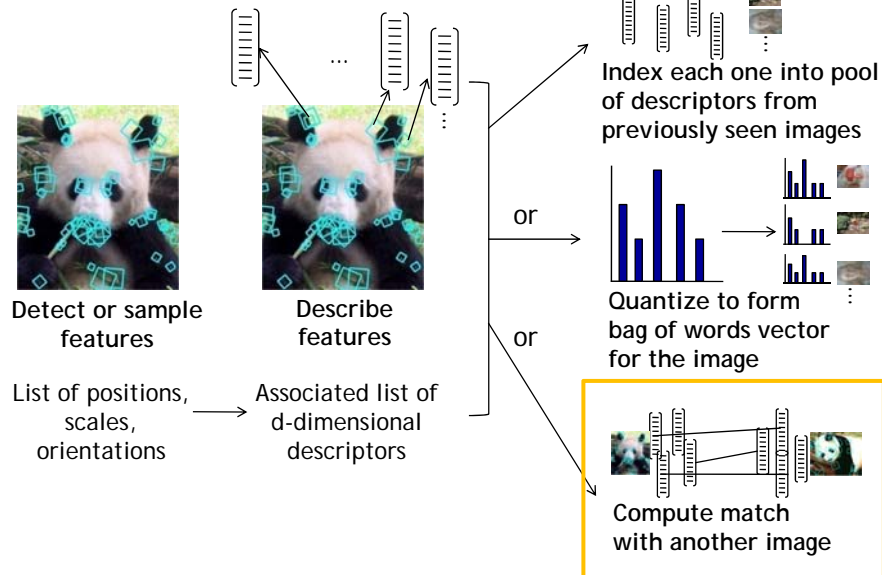
Source: Lana Lazebnik

### Bags of words: pros and cons

- + flexible to geometry / deformations / viewpoint
- + compact summary of image content
- + provides vector representation for sets
- + has yielded good recognition results in practice
- basic model ignores geometry - must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- interest points or sampling: no guarantee to capture object-level parts
- optimal vocabulary formation remains unclear



## Basic flow



## Local feature correspondences

- The matching between sets of local features helps to establish overall similarity between objects or shapes.
- Assigned matches also useful for localization



Shape context  
[Belongie & Malik 2001]



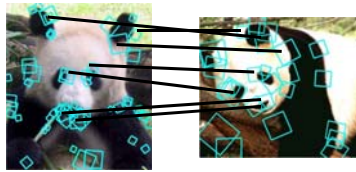
Low-distortion matching [Berg & Malik 2005]



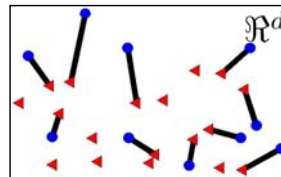
Match kernel  
[Wallraven, Caputo & Graf 2003]

## Local feature correspondences

- Least cost match: minimize total cost between matched points



$$\mathbf{X} = \{\vec{x}_1, \dots, \vec{x}_m\} \quad \mathbf{Y} = \{\vec{y}_1, \dots, \vec{y}_n\}$$

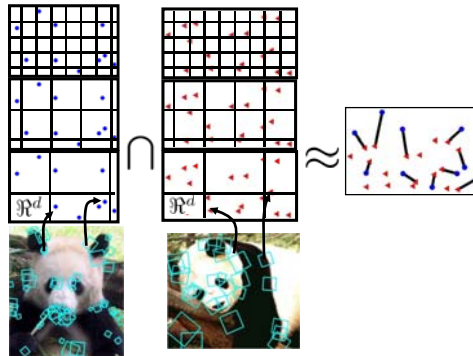


$$\min_{\pi: X \rightarrow Y} \sum_{x_i \in X} \|x_i - \pi(x_i)\|$$

- Least cost *partial* match: match all of smaller set to some portion of larger set.

## Pyramid match kernel (PMK)

- Optimal matching expensive relative to number of features per image ( $m$ ).
- PMK is approximate partial match for efficient discriminative learning from sets of local features.

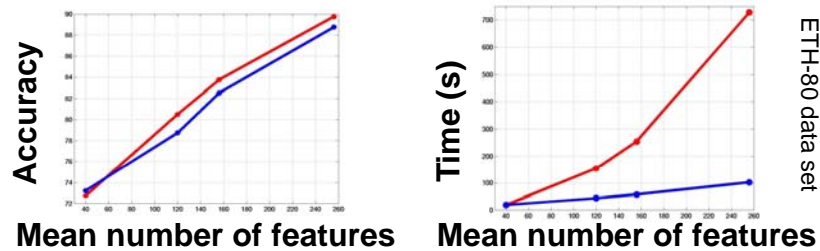


Optimal match:  $O(m^3)$   
 Greedy match:  $O(m^2 \log m)$   
**Pyramid match:  $O(m)$**

[Grauman & Darrell, ICCV 2005]

## Pyramid match kernel

- Forms a Mercer kernel -> allows classification with SVMs, use of other kernel methods
- Bounded error relative to optimal partial match
- Linear time -> efficient learning with large feature sets



Match [Wallraven et al.]

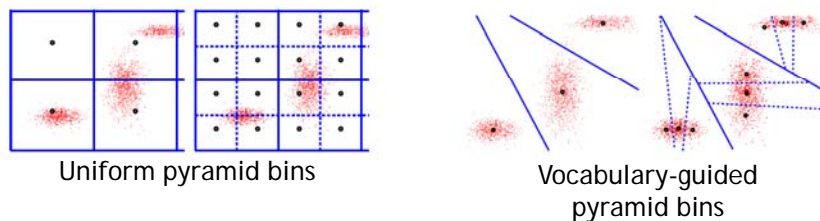
 $O(m^2)$ 

Pyramid match

 $O(m)$ 

## Pyramid match kernel

- Forms a Mercer kernel -> allows classification with SVMs, use of other kernel methods
- Bounded error relative to optimal partial match
- Linear time -> efficient learning with large feature sets
- Use data-dependent pyramid partitions for high-d feature spaces



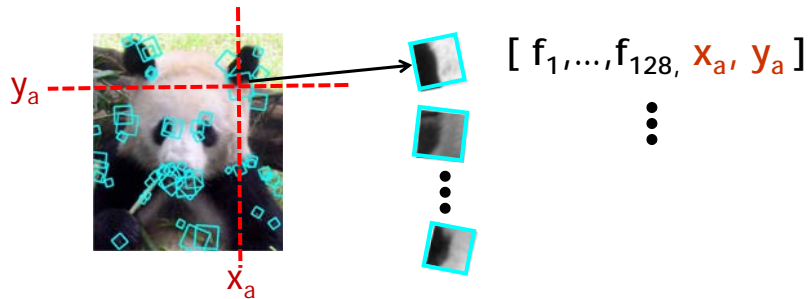
Uniform pyramid bins

Vocabulary-guided pyramid bins

Code for PMK: <http://people.csail.mit.edu/jjl/libpmk/>

## Matching smoothness & local geometry

- Solving for linear assignment means (non-overlapping) features can be matched independently, ignoring relative geometry (as in bag of words model).
- One alternative: simply expand feature vectors to include spatial information *before* matching.

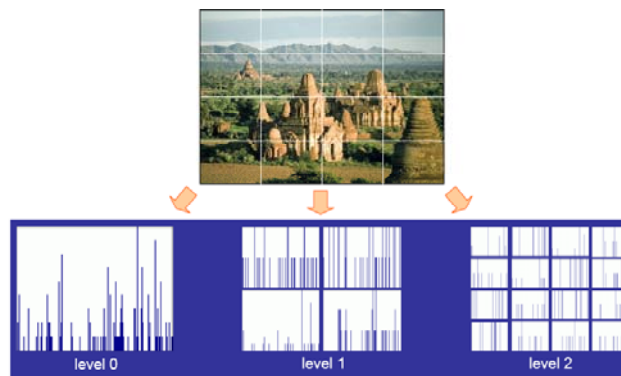


K. Grauman, B. Leibe

71

## Spatial pyramid match kernel

- First quantize descriptors into words, then do one pyramid match *per word* in image coordinate space.

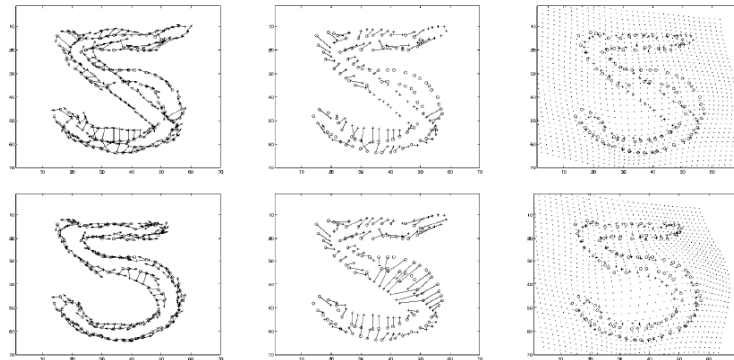


Lazebnik, Schmid &amp; Ponce, CVPR 2006

K. Grauman, B. Leibe

## Matching smoothness & local geometry

- Use correspondence to estimate parameterized transformation, regularize to enforce smoothness



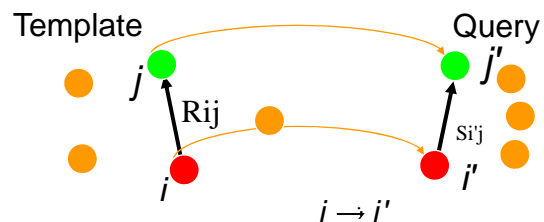
Shape context matching [Belongie, Malik, & Puzicha 2001]

Code: [http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc\\_digits.html](http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc_digits.html)

K. Grauman, B. Leibe

## Matching smoothness & local geometry

- Let matching cost include term to penalize distortion between pairs of matched features.



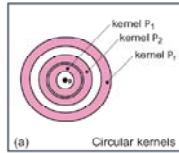
Approximate for efficient solutions: Berg & Malik, CVPR 2005;  
Leordeanu & Hebert, ICCV 2005

Figure credit: Alex Berg

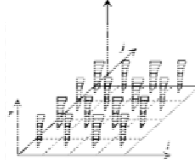
K. Grauman, B. Leibe

## Matching smoothness & local geometry

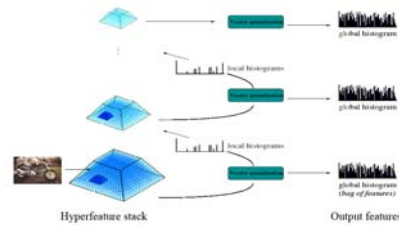
- Compare “semi-local” features: consider configurations or neighborhoods and co-occurrence relationships



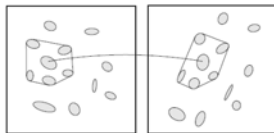
Correlograms of visual words  
[Savarese, Winn, & Criminisi, CVPR 2006]



Proximity distribution kernel  
[Ling & Soatto, ICCV 2007]



Hyperfeatures: Agarwal & Triggs, ECCV 2006]



Feature neighborhoods [Sivic & Zisserman, CVPR 2004]

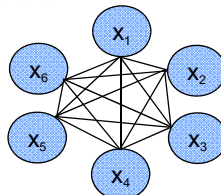
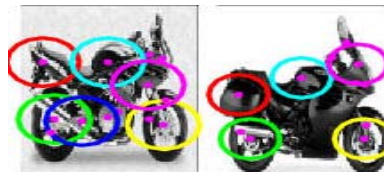
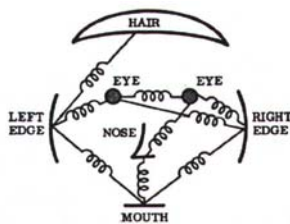


Tiled neighborhood [Quack, Ferrari, Leibe, van Gool ICCV 2007]

K. Grauman, B. Leibe

## Matching smoothness & local geometry

- Learn or provide explicit object-specific shape model  
[Next week: part-based models]



## Distance/metric/kernel learning



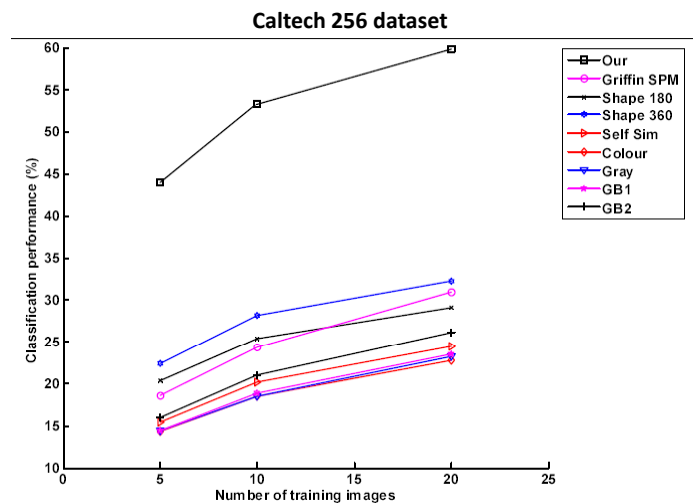
- Exploit partially labeled data and/or (dis)similarity constraints to construct more useful distance function
- Number of existing techniques

## Distance/metric/kernel learning

- “Multiple kernel learning”: Optimize weights on kernel matrices, where each matrix is from a different feature type or similarity measure. “Align” to the optimal kernel matrix.
  - [e.g. Varma & Ray ICCV 2007, Bosch et al. CIVR 2007, Kumar & Smichisescu ICCV 2007]
- Example-based distance learning: Optimize weights on each feature within a training image
  - [Frome et al. ICCV 2007]
- Learn metric based on similarity / in-class constraints
  - Often Mahalanobis distances [e.g. Hertz et al. CVPR 2004, Kumar et al. ICCV 2007, Jain et al. CVPR 2008]



## Example: impact of kernel combination



Varma and Ray, ICCV 2007

## Example Applications: Local Feature Matching

### Sony Aibo (Evolution Robotics)

#### SIFT usage

- Recognize docking station
- Communicate with visual cards

#### Other uses

- Place recognition
- Loop closure in SLAM



Slide credit: David Lowe

## Example Applications: Local Feature Matching



### Mobile tourist guide

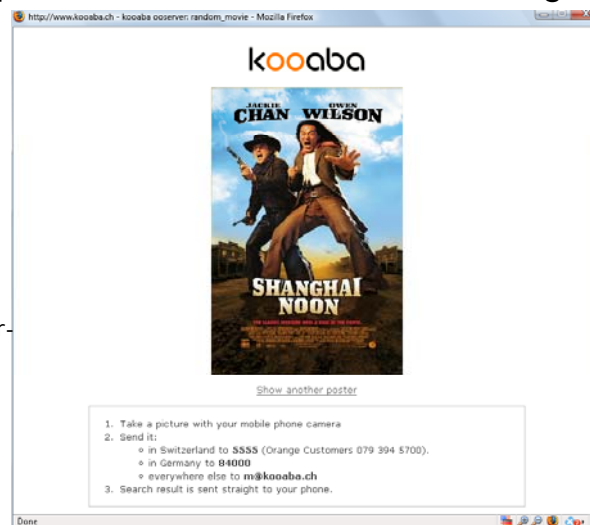
- Self-localization
- Object/building recognition
- Photo/video augmentation

[Quack, Leibe, Van Gool, CIVR'08]

## Example Applications: Local Feature Matching

50'000 movie posters indexed

Query-by-image from mobile phone available in Switzerland



[http://www.kooaba.com/en/products\\_engine.html#](http://www.kooaba.com/en/products_engine.html#)

K. Grauman, B. Leibe

## Summary

- Local features are a useful, flexible representation
  - Invariance properties - typically built into the descriptor
  - Distinctive, especially helpful for identifying specific textured objects
  - Breaking image into regions/parts gives tolerance to occlusions and clutter
  - Mapping to visual words forms discrete tokens from image regions
- Efficient methods available for
  - Indexing patches or regions
  - Comparing distributions of visual words
  - Matching features